DESIGN AND IMPLEMENTATION OF A DECISION SUPPORT SYSTEM FOR I. I. T. KANPUR RESEARCH & DEVELOPMENT CELL

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY

By

ANIL KUMAR SINHA

to the

INDUSTRIAL AND MANAGEMENT ENGINEERING PROGRAMME

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

JULY, 1984

Dedicated to

My dear Parents

CERTIFICATE

This is to certify that this work entitled, 'DESIGN AND IMPLEMENTATION OF A DECISION SUPPORT SYSTEM FOR IIT KANPUR RESEARCH & DEVELOPMENT CELL', by Mr.Anil Kumar Sinha, has been carried out under my supervision and that it has not been submitted elsewhere for the award of a degree.

July, 1984.

(S.Sadagopan)
Assistant Professor
Industrial & Management Engg.
Indian Institute of Technology
Kanpur-208016

ACKNOWLEDGEMENTS

I feel deeply indebted to Dr.S.Sadagopan for his dynamic inspiration and guidance provided throughout the course of preparation of this thesis.

I would not be fair if I don't put on record, my sense of appreciation for the valuable help provided to me by my friends, especially Mr.B.S.R.Sarma of Computer Science Department.

Mr.U.S.Misra and Mr.Budhi Ram Kandiyal have been extremely helpful, in their usual efficient way, in typing the text and the cyclostyling work respectively. My sincere thanks to them.

Last, but not the least, I express my appreciation of all the patience and supporting attitude of my wife Vijaya.

-ANIL KUMAR SINHA

TABLE OF CONTENTS

ABSTRACT

This thesis describes the design and test implementation on sample data, of a computerized Information System for Decision Support for the Research and Development Cell at IIT-Kanpur, developed on the DEC-10 computer.

The emphasis in the system design has been on the development of a human/computer interface, using an inter-linked system of FORTRAN programs and MIC (MACRO Interpreted Commands) facility, to enhance the direct interaction of the manager with the system. This interface is built around and linked to the IQL (Interactive Query Language) package for information management, available on the DEC-10 computer system.

The system, using a Question-Answer type of interface, enables the user to retrieve information from the data bank, using either a set of pre-designed queries or by creating his own query.

CHAPTER I

INTRODUCTION

1.1 Decision Support Systems:

The early work, done for a computerized support system for managers, concerned itself with a Management Information System (MIS). The term MIS implies that the information system is for the managers. However, many managers felt that it was not so. They claimed to be uninfluenced by computer based systems although they willingly granted that the organization under them had been considerably affected [6]. Even the most simplified information systems required some programming skill and imposed a structure on the operating methods. At best, they provided the manager with an indirect means of getting reports generated through somebody else. Their direct interaction with the system was almost negligible. The move towards the Decision Support Systems (DSS) has come about to overcome this deficiency through designing a system where the direct interaction of the manager with the computer is increased and the operating method made easy.

The DSS is defined as "a coherent system of computer based technology (hardware, software and supporting documentation) used as an aid to decision making in semi-structured decision tasks" [6]. The stress is on

supporting rather than replacing decision making.

A task is said to be unstructured when —

i) Objectives are ambiguous and non-operational or objectives are relatively operational but numerous and conflicting.

ii) It is difficult to determine the cause of changes in decision outcomes and to predict the effect on decision outcomes of the actions taken.

iii) It is uncertain what actions taken by the decision maker might affect decision outcomes.

The focus here is on objectives. The unstructured tasks are unable to connect the user actions with favourable decision outcomes in any deterministic way. In other words, eventhough it is known that some action is required to obtain and use information for decision making, we cannot prove that a given set of actions is essentially linked to the final quality of the decision outcome.

The "Decision Support System" (DSS) refers to an interactive system which has access to the analytical power and the active database of a computer. All this is needed to fulfil the main objective of helping the decision maker in solving semi-structured problems [2].

The success of an interactive computer system depends to a large extent on a good human/computer interface.

It is all the more important for a DSS that the user and the system interact in a conversational mode to supplement the user's judgement with the analytical power of a computer [5]. Besides, since the users of a DSS are most often not technical specialists, the interface should not impose any structure with which the user feels uncomfortable.

1.2 The Scope of the Thesis:

This thesis describes the design and implementation of a Decision Support System.

The activities of the Research & Development Cell at IIT, Kanpur have been taken in this exercise to serve as a model for this information system.

The main emphasis is on the development of an easy-to-use interface so that the system is accessible to any casual user with no significant knowledge of computer programming.

1.3 Organisation of the Thesis:

Chapter II discusses the general background material required for understanding the activities of the R&D cell and the scope for a Decision Support System (DSS) in it.

Chapter III deals with the system design, going into the aspects of system structure, the use of a DSS generator, the adaptability of Interactive Query Language (IQL) package,

available on DEC-10 computer, as a DSS generator and the development of the main controller of the interface using MIC (MACRO INTERPRETED COMMANDS).

Chapter IV goes into the implementation details, giving an account of the various programs developed and their functioning.

Finally Chapter V is for conclusion and it also discusses the scope for future extension in this system.

CHAPTER II

SYSTEM DESCRIPTION

2.1 Research & Development Cell:

One of the strengths and objectives of the IIT,Kanpur is the training of the minds of young students, the training of minds to analyse the problems and find rational solutions to them. The research and development activities are the two most important activities of the Institute in fulfilling this objective. The R&D activities broadly consist of:

1. Sponsored Research Projects
2. Development and Consultation Projects
3. Intensive Courses in Advanced Topics for other Educational Institutions
4. Refresher Courses to the Industry
5. Specialized Courses for a type of Industry or for an Industry
6. In-home courses for Industry
7. High Level In-house Industrial Consultation during Vacation
8. Interaction with AIEI on Educational and R&D Programmes
9. Seminars at IIT-Kanpur by visiting specialists
10. Symposia and Conferences
11. Writing books and Monograms

12. Institute Lecture Series
13. Entrepreneurship Training
14. Patents.

For the purpose of the present system, let us concentrate on the area of research projects. As seen in the above list, these are of two types.

2.1.1 Sponsored Research Projects:

A group of faculty members or a member of the faculty can propose research projects of special interest, either to them or of special need of the Nation. Sometimes these projects are proposed by the Governmental Departments.

Research projects are generally long term projects whose findings and results have to be followed up by the sponsoring agency to put to proper application. Very often, the results of the sponsored projects are in a form which need further refinements before they could be applied into practice.

2.1.2 Consultation Project:

The consultation projects are highly goal oriented and time-bound. A project is usually originated by an agency for a specific problem and referred to the Director or to the Dean of Research and Development or to the Head of a Department or to a Faculty Member. In many cases,

the industry or the organization identifies the faculty member even at the very beginning. Member of the faculty have their own specializations in the areas of product development, process, design, management or computer aided design etc. Most problems referred by the industry are complex and require specialized knowledge of the faculty members. These projects are time-bound and have a specific purpose of application. The results must be directly usable in the Industry. There is a great responsibility and challenge to the faculty members in undertaking such projects. A considerable pressure exists to deliver the results on time.

To meet the continuous demand of the faculty time, the dead lines and to overcome the constraints, the faculty and the staff members are given considerable flexibility and financial incentives. Depending upon the nature of the project, there are norms provided by the Institute to utilize the consultation fee for actual expenses, travel, honorarium, Institute overheads and R&D fund.

The main weakness in this area is found to be in meeting the deadlines of projects which may cause problems both ways.

2.1.3 Industrial Interaction:

The faculty has been arranging advanced and refresher courses for Industry as well as for other Educational Institutions. During the period of organising the courses, there is a considerable interaction, exchange of ideas, initiation of special and new courses or projects. It is felt that opportunities for improvements, better organizational and administrative support exist.

2.2 Need for an Information System and its Scope:

As inferred from the above discussion, an effective and easy-to-use information system for Decision Support could help the R&D cell in better monitoring of sponsored projects, where the time pressure is high. With an effective means of interactively obtaining information regarding projects with a particular faculty member or department or for a sponsor or in a given year etc. could help the administration of project activities by reminding the concerned personnel, timely communication with the sponsoring agencies, anticipating load on a particular faculty member or department in a given year etc. It could also help decision making with regard to better distribution of load in appropriate cases.

In case of new projects being referred to by Industry, the Dean R&D could immediately check the appropriate faculty members, specializing in the concerned field, their

current workload etc. and make decisions regarding the project allotment etc.

Another use for this DSS could be for demonstrating to the personnel from Industry, the expertise available, projects in progress etc. which could lead them to think about awarding new projects to the Institute. The industry people could be persuaded to use this system themselves. This would be easily possible because of the easy-to-use, non-programmer interface provided for this DSS. The opportunities for such interaction would be amply provided during the visits from Industry for the courses etc. organised by the R&D cell. Some of the industry personnel might even be tempted to have an appropriate DSS designed by the Institute for their own industries.

In view of the above, since the expected users would be varied eg. Dean R&D, his staff, industry people and casual visitors etc., the absolutely essential part of an effective DSS would be its ease of usage provided by an appropriate interface.

Besides, the DSS would also be useful for routine report generations which R&D needs from time to time.

2.3 Cost/Benefit:

General studies of decision making have indicated the potential benefits of a computer support for decision making. These potential benefits can be divided into two

categories : displaced cost and added value. Displaced cost results from reduced costs for data gathering, computation, and data presentation in support of decision making. Added value results from investigating more alternatives, doing more sophisticated analyses, making quicker decisions etc. often, it is difficult to identify the added value because it does not occur on a routine basis. Hence it is difficult to quantify the benefits thus accruing although small improvements in decision making can result in large long term benefits.

With the steep decline in hardware prices and easy availability of powerful computers, the benefits of an interactive Decision Support System would soon offset the cost of the system itself.

CHAPTER III

SYSTEM DESIGN

3.1 Introduction:

The strategy for design and development of a DSS is one of evolution over a period of time. DSS deal with semi-structured problems and hence the design and implementation are inseparable and evolutionary. As a result of this the initial design will, most often, prove to be incomplete. This is all the more true as the decision makers learn more about the problem and problem-solving process through analysis, implementation and usage of the DSS. The perception of various situations and problems changes with time and usage. Thus there is no clear break between design and implementation. Similarly, there may be no precise end to implementation; evolution of ideas, the user's ongoing learning, and a shifting environment lead to new adjustments and developments.

The DSS designed and implemented in this present exercise is a sort of an initial design. The structure is modular and consists of several small and separate parts which are linked together. This is to enable ease of extension and modification as the experience grows. This is in keeping with the evolutionary or adaptive approach to designing a DSS, as suggested by a number of experts in this field.

3.2 Objectives and General Framework:

The main purpose of the DSS designed and implemented in the present exercise, is to serve as a model and an initial design. Hence, the following objectives were selected to serve as the main guide line —

1. To obtain information regarding sponsored consultation projects for any given key eg. Department, Sponsor, Faculty member, Year, Level of grant or Research field. The information could be a simple list and/or various aggregates.

2. To obtain information regarding the faculty members and their respective specializations. The key could be the name of a faculty member or a given research field.

Two separate files containing records with information pertaining to projects and faculty specializations were created. Since the number of records is not very large and the idea is to have more or less uniform search time with respect to any field in the record, a sequential file structure was selected.

3.3 DSS Generator:

M.Alavi and H.A.Napier [1] point out that, in an adaptive design framework, the technical system is the hardware/software provided to the user. A technical system is configured from a DSS generator. A generator is a package which provides a set of capabilities to build a specific DSS easily and quickly [8] eg. report generation, inquiry etc.

The package used as a DSS generator in this present exercise is the 'Interactive Query Language (IQL)'. The IQL has very robust, powerful and user-friendly facilities for creating, updating and inquiring files with any structure eg. sequential, indexed sequential or data base. It also has extremely good report generation and aggregation facilities. Many features of IQL are similar to the features of COBOL. However, IQL is not a programming language. IQL is a general-purpose query language with the ability to interactively interrogate data files and generate reports based on that data. Rather than write a separate COBOL program for each application, we can use IQL.

According to the IQL manual the Interactive Query Language is an information retrieval and reporting system. It can be used to:

1) Create data dictionaries of the data files
2) Read a specific piece of information
3) Change a specific piece of information

4) Update an entire file

5) Copy selected portions of data to a new file

6) Write reports based on the data.

When working with a file, IQL can be used to set up the format of a report, sort records, summarize data, compute new data and write new data in the file.

One basic advantage of using IQL is that it can be used equally well by experienced computer professionals or by people with much less experience.

The two key elements of the Interactive Query Language (IQL) are queries and dictionaries. A query is a set of IQL statements that are used to manipulate data and generate reports. A dictionary is an exact description of a data file. A dictionary is similar to a File Description (FD) in a COBOL Data division. Both contain storage and access information such as item definitions, editing pictures, usage declarations, blocking factors, and definitions of ISAM or DBMS keys. In addition, the dictionary holds a 'title' for each item. The report generator uses this title when column headings are needed.

3.4 General Strategy for File Manipulation:

As discussed above, the data records containing information regarding sponsored projects and faculty-specializations are created in two separate sequential

files i.e. one file for projects and another for specializations. The specific file structures are described later.

The strategy is to write a number of IQL queries to extract information from these files with reference to different keys. Each query is stored in a different file. The main control program of the interactive interface takes the appropriate key values and output requirements from the user and selects appropriate query file, executes it and produces the desired output report.

A query is a collection of source statements written like English language sentences that IQL uses to generate one or more reports. The easy structure of queries makes it very easy to add new queries as the requirements grow and become more specific in the light of experience with the initial DSS.

3.5 Design of the User Interface for DSS:

3.5.1 Introduction:

The initial users of computer system were a small group, basically consisting of its designers in the laboratory. Each person was aware of the machine capabilities and information necessary for effective use of the system. The user could literally visualize the flow of data through physical components such as AND and OR gates. As computers evolved — people learned the higher level

abstractions eg. registers, words, memory locations and computer logical instructions. These enabled a user to develop computer programs even without knowing the internal, physical details of the machine. However, the use of these abstractions required specialized skills on the part of the user. Thus, in these earlier stages, the designers and programmers were 'committed users', as a result of personal interest or job requirements [3].

Nowadays, we find a new class of people becoming interested in using computers. The 'Discretionary Users' work with computers by choice. For example, managers have access to many sources of information and have available, many tools for data manipulation. In order to be able to capture their interest and to support their application of judgement in deriving results, we must provide presentations which are directly relevant to managers and give them active control over interaction.

Display terminals, which show information on a screen and respond to user input at a conversational pace, can provide a level of service attractive to the discretionary users. Immediate feedback gives a sense of presence, user involvement and participation.

This new level of support for discretionary users, made possible by the display terminals has become technically and economically feasible. However, it still requires the

users to be knowledgeable about underlying computer-oriented details. These systems are suitable for people committed to computer use eg. designers, programmers etc. But discretionary users are not computer professionals. Thus direct, interactive use of computer power by managers depends upon the designer ability to provide interfaces that these people are willing to use.

3.5.2 Relevance of Interface for DSS:

DSS provide high-level operations for retrieving data, generating reports and alternatives etc. The data relevant to the solution of problems may already exist in the computerized data base. However, the data are rarely available in the required form, hence the need for interactive support to the manager using the system.

3.6 A Schematic for the User-Computer Interface:

Figure 3.1 shows a schematic representation of a user interface. Information presented at a terminal passes across the interface. The "processor" within the user operates on the observed content in order to formulate an action plan. The plan must then be translated into a set of actions accepted by the terminal. User understanding of the displayed information is influenced both by the work being done and his knowledge of the computer system.

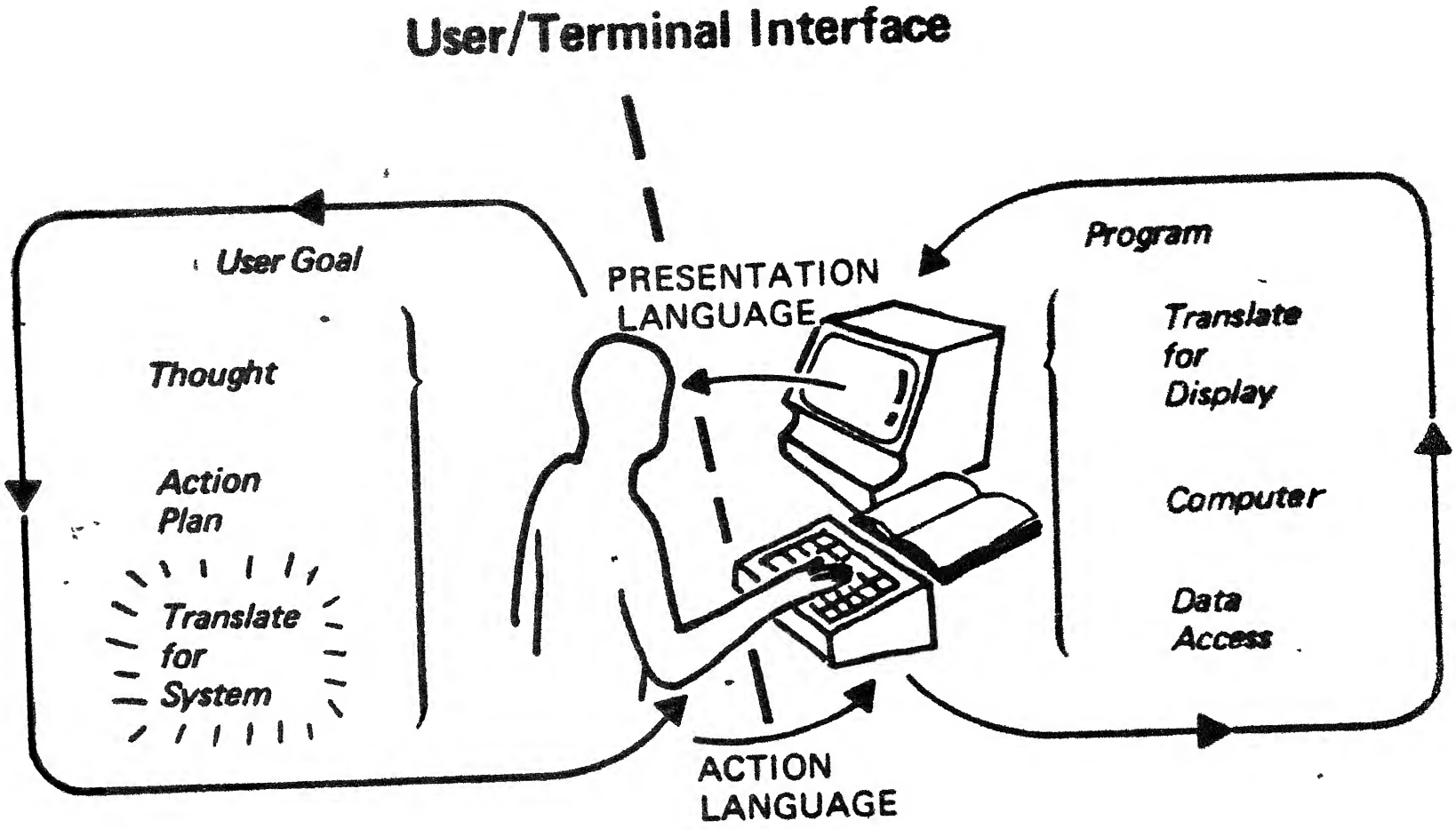

Fig. 3.1 Schematic Representation of a User Interface.

In Figure 3.1 "thought" refers to user interpretation of the displayed data. "Action Plan" refers to input the user plans. "Translate for System" refers to any steps the user must take to transform the direct actions into actions that the system will accept. The computer processor completes the loop by interpreting the user actions, accessing stored data where necessary, calculating needed values, and translating the data into a form understandable to the user. If the user does not need to know the internal structure in order to use the system, then the system implementation need not concern the user.

3.7 Specification of the User Interface:

A user interface design that is very common in DSS employing line-at-a-time terminals, is the "Question-Answer" (Q/A) design. The Q/A interface design tends to be most successful with inexperienced or infrequent users who are unfamiliar with the problem to be solved. The main disadvantage of a Q/A design is that it leads to awkware usage patterns if, during a dialogue, user needs to modify answers to previous questions.

With Q/A design, the DSS asks the user a question (possibly multiple choice), the user answers the question and so on, until the DSS produces the answers needed to support the decision. Normally, a Q/A interface would use

"natural" language and may determine the next question based on the answer(s) to the previous question(s). If the DSS cannot "understand" an answer, or needs additional information, clarification questions may be asked.

To make the user interface more effective the Q/A interface could be combined with a MENU interface for routine interactions. The MENU interface allows the user to select, at the MENU level, one of the many main branches available to go down in the program structure of the DSS.

It has been shown [4] that the largest costs in developing the software for a DSS is incurred on development and maintenance of software which implements the user interface. As an example, about 60% of the code written for DSS GADS, is code for user interface.

3.8 Modular Structure of User Interface Code:

Modular means "standard design", so modularization means, to make standard. Modularization is a common programming technique, and it usually involves dividing the program into well defined, self-contained, small units with one entry and one exit.

There is a high probability that the user interface will need to be modified during the life cycle of the DSS. Hence, the code for the user interface should be in program units (modules) which are separate from the remaining code.

The code that reads and writes the data to and from the terminal should also be in modules separate from the code that defines the output and input formats for a specific interface.

3.9 Conclusion:

An effective user interface does not guarantee the success of a DSS, but it is a necessary ingredient. There is little research that indicates the relative importance of the interface with respect to many other factors which influence the success of the DSS. Two facts, however, do indicate its importance—

1) The code for the user interface often represents the largest percentage of the total code in a DSS and is, most often modified code [6].

2) Communication is known to be an important ingredient in effective decision making and the user interface in a DSS is the means through which the users communicate with the DSS.

With sufficient experimental data it should be possible to evolve a user interface so as to reduce cost and increase effectiveness.

CHAPTER IV

DETAILS OF IMPLEMENTATION

In this Chapter we discuss the specific implementation details of the system including the various programs, their structure and the mode of linking them. The basic implementation strategy is as follows —

4.1 Elements of the System Code:

The code for the DSS consists of three parts namely

1) Queries written in Interactive Query Language (IQL) to extract the requisite information from the data files and generate the required reports.

2) The control programs written in higher level programming language FORTRAN that take the input from the user interactively, using a question-and-answer mode and transmit it through the interface to the appropriate segment of the program. The input could be regarding the selection of the next step and/or regarding the value of a key to be used for the search through the sequential data files.

3) The main controller of the DSS is designed using the MIC (MACRO INTERPRETED COMMANDS) package. This is a very powerful package which

decides about the sequence of monitor commands to be given to the DEC-10 computer system, used for this DSS. This sequence can be governed by IF conditions depending upon the values of certain parameters which are changed during the execution of the programs, depending upon the inputs given by the user.

4.2 Structure of Data Files:

File processing involves a number of activities. One of the most basic is information retrieval. This retrieval could be based on one or more keys. In order to change the contents of the file to reflect new information, a different type of processing-updating is required [7] .

Updating a file consists of three activities: adding records, deleting records, and modifying records.

IQL provides very powerful and easy-to-use features for creating and updating the data files. For any data file, first a dictionary defining the file and record format is created and then, using this dictionary, the data file can be created or updated.

Since the DSS here may query the file using any field using any field of the record as key, the sequential file structure is best suited.

This DSS concerns itself with information regarding sponsored consultation projects and specializations of faculty members. Therefore, we have two sequential files for data as described below.

4.3 File Description:

1) File: PROJ.SEQ (for consultation projects)

The record structure is as follows—

1) Serial Number
2) Research Field
3) Research Topic
4) Department Initials eg. ME, CS, METE etc.
5) Full Department Name
6) Last name of First Faculty Member
7) Initials of First Faculty Member
8) Last name of Second Faculty Member
9) Initials of Second Faculty Member
10) Initials of Sponsoring Agency
11) Full Name of Sponsoring Agency
12) City where Sponsoring Agency is located
13) Starting year of the Project
14) Ending year of the Project
15) Grant amount.

2) File: SPEC.SEQ (for faculty specializations)

Following fields are there in each record—

1) Serial Number
2) Last name of Faculty Member
3) Initials of Faculty Member
4) Department Initials
5) Full Department Name
6) First field of Specialization
7) Second field of Specialization
8) Third field of Specialization
9) Fourth field of Specialization
10) Fifth field of Specialization.

4.4 Information Retrieval:

4.4.1 First Phase of Implementation:

Now, a record could be retrieved with any of the fields as a key using IQL queries. Presently we have the following searches possible—

1) PROJECTS can be searched
   a) Departmentwise
   b) Sponsorwise
   c) Name of Facultywise
   d) Amountwise i.e. projects with grant above a given amount

e) Yearwise i.e. projects in any given year

and f) Field of Researchwise.

2) SPECIALIZATIONS could be searched

a) Name of Facultywise

and b) Field of Specializationwise.

For each of the above described searches a separate query in Interactive Query Language is written and stored in a separate MIC file.

Search based on any of the keys could be for specific values of other keys also eg. a departmentwise list of projects could also be searched for a specific sponsoring agency, a specific faculty member and so on. The strategy for achieving this kind of a search is that all records with the first key matching the given value are copied into another file and subsequent searches are carried in this new file. The advantage of this is that the same IQL query for the search based on a particular key could be used irrespective of whether the key is the first one supplied by the user or any subsequent one. All possible combinations of the keys in any order are thus possible.

Depending upon the input from the user the control programs of the interface execute the MIC file containing the appropriate IQL query, supplying it with all necessary information through the parameter transmitting facilty of the MIC and produce the required report.

4.4.2 Second Phase of Implementation:

This phase of implementation, which has been partly implemented, involves two stages which are as follows—

1) Addition of more queries:

The choice of IQL package as a DSS generator here, has been made due to the power of IQL for writing with relative ease, reasonably complex queries for performing various combinations of sorting, searching, aggregation of data, keeping track of counts of various items simultaneously, report formatting, copying records to different files, creating new files, making a join of two or more files etc.

This makes the job of extending the present DSS extremely simple. The modular structure of the control program and ease of designing queries using IQL puts enormous power in the hands of the designer to keep on adding new queries to the system to make the DSS more and more flexible and versatile.

Presently the queries, that have been written, are rather simple ones using essentially a tabular search in single data files. This is basically because the idea in this exercise has been to build a nucleus DSS which could grow into a reasonably large system with passage of time. Following are a few queries which could be added on to the system straight away for the existing data files.

i) Get the number of projects for each department or each sponsor or each faculty member for a given period of time or above a given grant level etc.

ii) Get the total grant amount apportioned to a given faculty member; a given department etc.

iii) Get a list of projects on which the faculty members, having a given field of specialization, are working.

Unlike other queries, where the search is within a table, here, join of the two data files or tables, namely PROJ.SEQ and SPEC.SEQ, has to be made to accomplish the search.

2) Implementing the New Query Module:

This is the stage where the transition to a genuinely versatile and flexible Decision Support System starts. So far, the system has been bound into a relatively fixed set of pre-determined queries. With the implementation of this module, the user is provided an opportunity to create his own IQL queries without going through the rigours of learning IQL, because, in a managerial environment, the user is reluctant to use anything which imposes any structure. With this module, the user indicates his requirement of operations through a Question-Answer interface and the module generates a corresponding IQL query which is run by

the main control module and the output displayed.

This module has currently been implemented for a few basic functions of IQL and can be made more versatile in the next phase.

4.5 Control Programs of the DSS Interface:

All the control programs of this conversational interface are coded in the higher level programming language FORTRAN.

The main program is a MENU which provides the alternatives to the user regarding which file he wants to search and what is the next step that he wants to take, i.e. which key does he want to base his search on. Depending upon his responses input here the next program to be run by the MIC controller is selected. There are several small and independent FORTRAN programs, one for each route of search i.e. each key available for information retrieval, written in separate files. As per the route selected by the user in the MENU program, the appropriate program is run which, in a conversational mode, takes the user's responses regarding the value of the key he has selected in the MENU program and also regarding the next step that he wants to take. He has an option to inspect the available key values at this step.

Once the user completes his input to this step, the MIC controller uses the inputs provided by the user and runs the appropriate IQL query and generates the required report in a pre-defined report format. IQL provides very powerful report formatting facilities. The inputs to the IQL query eg. the data file to be used and key values are transmitted through the parameter transmitting facility of MIC via the MIC file containing the query.

If the key value is correctly specified, then the appropriate records are sorted and the report generated. However, if the key value is incorrect or does not match for any record in the file, then the program gives an appropriate message and instructs the main MIC controller to run another IQL query which gives a list of all available values for that field for the records contained in the current input file. After this the controller sends the control back to re-run the previous input program to ask for a fresh key value from the user.

After a report has been generated, the controller runs a program which gives the user an option to retain a print file for this report or otherwise. The user can save a print file under a name specified by him.

At the next step the controller runs a program which ascertains the user's intention regarding either exiting from the system or going back to the MENU for allowing

the user to take a re-run of the system.

The above description gives a general outline of the strategy that the DSS designed in the present exercise utilizes to converse with the user, ascertain his requirements and generate the reports desired by him. The flow of control is depicted in a schematic form in Figure 4.1.

4.6 List of Programs and their Functions:

Following is a list of actual programs and a brief description of the functions performed by them.

4.6.1 FORTRAN Programs

1) MENU:

This is the main interface program which asks the user to indicate whether he wants the information for Projects or Specializations or wants to exit. Then, based on the route he indicates, further questions are asked regarding the key he wants to base his search on etc. The user's responses are written in an output file through which they are transmitted to the MIC controller.

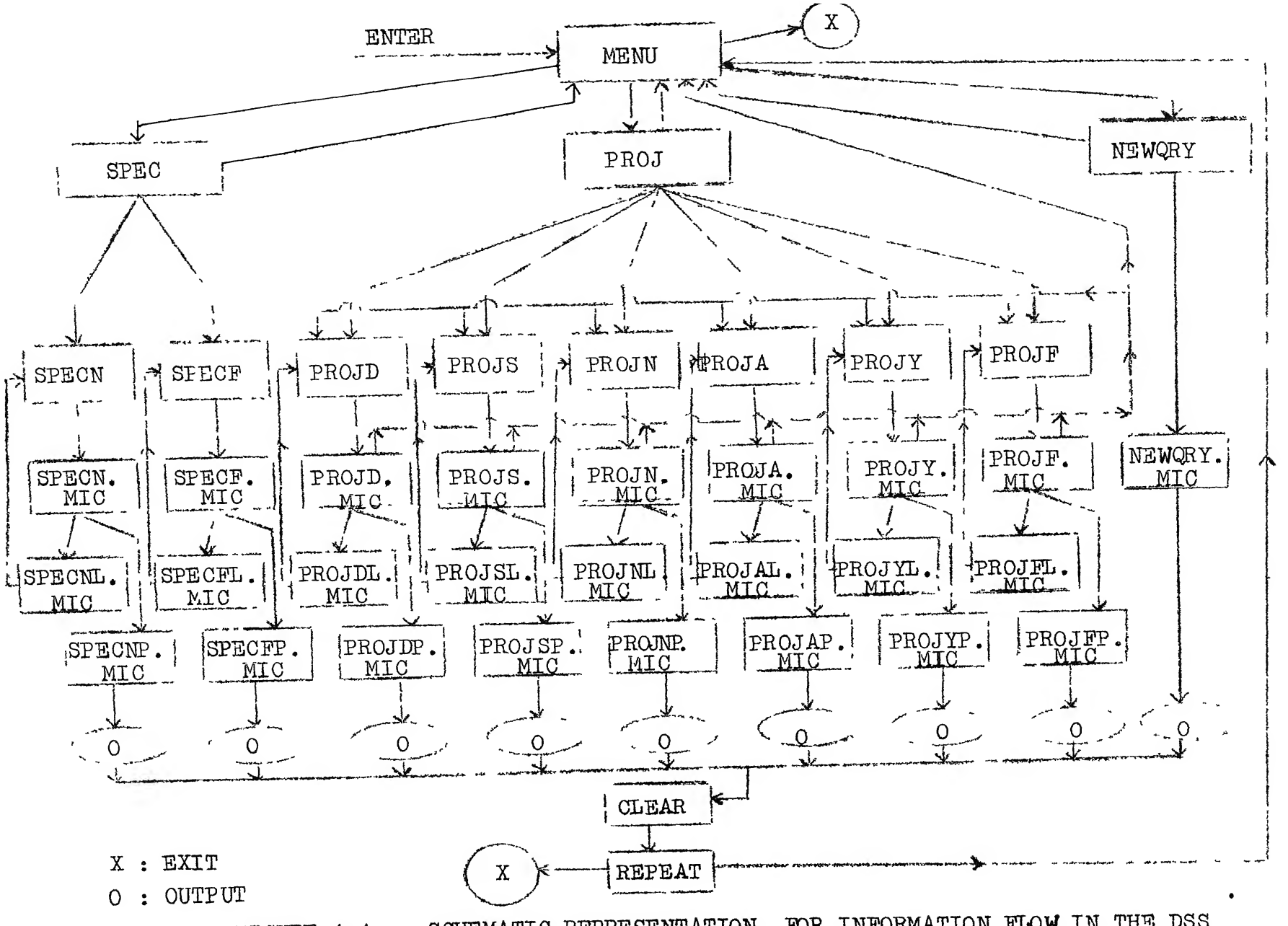

FIGURE 4.1 : SCHEMATIC REPRESENTATION FOR INFORMATION FLOW IN THE DSS

2) CLEAR:

After a query has been successfully run and a report generated, this program is run. It elicits a response from the user regarding whether he wants to retain a print file for his report and if so, under what name.

3) REPEAT:

After completion of one run, this program of the interface helps the user to decide whether he wants to quit or re-run the system.

4) FLASH:

This is a small program which clears the screen of the video terminal.

5) PROJD:

This input program is run if the user indicates to the MENU program that he wishes to search the data file for projects with "department" as the key. To this program the user gives his response regarding the value of the key (department) and the next step regarding any further sub-divisions of the search or otherwise.

6) PROJS:

Function similar to that of PROJD for sponsor name as the key.

7) PROJN:

Name of faculty as a key.

8) PROJA:

Grant amount as a key. The user is asked to specify a Lower limit for the grant amount.

9) PROJY:

Calendar year as a key. User specifies a particular year for which he wants all the projects running.

10) PROJF:

This program is run when the user has specified the field of research as a key.

11) SPECN:

This program of the interface is run when the user has specified in the MENU that he wants to search the data file for faculty - specializations based on faculty name as the key. The function is similar to that of the earlier mentioned in-put programs for Projects eg. PROJN etc.

12) SPECF:

Input program for search based on field of specialization as a key for faculty-specialization data file.

13) NEWQRY:

This program ascertains the user's requirements for writing of a new query using a Question-Answer format and passes this information on to the

corresponding MIC program which transforms it into an IQL query and runs it.

4.6.2 MIC Programs:

1. INTRO.MIC:

This is the main MIC (MACRO INTERPRETED COMMANDS) controller program. It is basically a set of monitor commands which are executed as per the user's response which are transmitted to this controller program from the interface programs through the parameter transmitting facility of MIC. Based on the parameter values, the control can flow to different sections of the MIC program.

Following MIC files are for searching through the data file for consultation projects.

2. PROJD.MIC:

This program file contains an IQL query for searching the data file for consultation projects, PROJ.SEQ and extracting the relevant records using department as a key. This file is executed by the main MIC controller and the query run as and when the user indicates it as his requirement.

3. PROJS.MIC:

This has a structure and function similar to PROJD.MIC except that the query here uses sponsor-name as a key.

4. PROJN.MIC:

This contains a query which uses the name of faculty member as key for searching in the data file PROJ.SEQ.

5. PROJA.MIC:

Contains a query using the grant amount as a key which extracts all records with grant money above an amount given by the user.

6. PROJY.MIC:

The query here uses the year as the key and extracts the records which have projects which would be running in the year given by the user as an input.

7. PROJF.MIC:

Records are searched and extracted based on research-field as key.

8. PROJDF.MIC & PROJSF.MIC:

These files contain IQL queries for generating full reports of all projects, departmentwise and sponsorwise respectively.

9. PROJDL.MIC, PPOJSL.MIC, PROJNL.MIC, PROJAL.MIC, PROJYL.MIC, PROJFL.MIC:

Each of these files contains an IQL query for extracting all the available key values in the input file for department, sponsor, faculty-name, grant amount, year and research-field respectively as key.

10. PROJDP.MIC, PROJSP.MIC, PROJNP.MIC, PROJAP.MIC, PROJYP.MIC, PROJFP.MIC:

These files contain queries for generating reports for the data finally extracted by previous query based on keys department, sponsor, name, amount, year and field of research respectively.

Following MIC files contain IQL queries for extracting records or performing other functions similar to that for the ones mentioned above for the data file for faculty-specializations, namely SPEC.SEQ.

11. SPECN.MIC, SPECF.MIC:

These files contain queries for extracting records from the data file SPEC.SEQ for a given value of the key faculty-name or field of specialization respectively.

12. SPECNL.MIC, SPECFL.MIC:

Queries for listing all available key values for faculty-name and specialization as key respectively.

13. SPECNP,MIC, SPECFP.MIC:

The queries stored here are used for generating the namewise or fieldwise list respectively for the records extracted from the data file by the previous queries.

14. NEWQRY.MIC:

This program generates a new IQL query according to the requirements indicated by the user and runs that query to output the desired report.

4.6.3 MACRO Program:

1. M.MAC:

This is a MACRO routine which is called from within each of the FORTRAN programs of the interface. It suppresses the normal system output at the end of execution of a FORTRAN program eg. the statements signifying the end, CPU time etc. This suppression is done to eliminate the display of unnecessary and unattractive information in the user interface.

CHAPTER V

CONCLUSIONS

In the exercise undertaken currently for developing a model for Decision Support System for project information has been designed and implemented on DEC-10 computer for the use of the Research and Development Cell of IIT, Kanpur.

As has been discussed earlier, a DSS does not intend to replace the manager or his tools. It merely is an extension to his techniques to support him in his decision making function.

"Behavioural scientists would approve of the emphasis in DSS on the user's needs, capabilities and procedures" [6]. The computer had not really been of major and direct value to managers in their problem solving. Information processing technology can be made useful to them, given an understanding of their needs.

It is felt to be essential to emphasise that the major points of focus in building a Decision Support System, which have been the chief consideration in this exercise, are as follows:

(1) An evolutionary approach to DSS design is necessary. DSS applications do not come tidily packaged with neat specifications. The emphasis therefore should be to get started with an initial design and then

build the system based on user's response and his growing experience.

(ii) In a DSS, the most important part of the system, which is vital to its success, is the user interface and this is where the major attention should be paid. The design of the dialogue is important in this respect.

The modular structure of the program makes it amenable to any extension or modification. If it is felt that some more types of queries based on the current data files are required or some new data files and queries have to be added, all that is required is a slight modification of the MENU program and writing of the new IQL queries in separate MIC files. All it needs to link these queries to the main controller is the naming of these files so that the output from the MENU program transfers the control to these files for execution.

Since the intention here has been to have a search based on any field of the record as key, the sequential file structure has been chosen, as it would be a bit too complex to have indexes for each field, besides requiring additional storage space. However, the DSS generator used here, namely IQL, can support Indexed sequential or Data base files also.

The choice of IQL as a generator is also important in the context of adaptability to future extensions. Since

the search procedure in this particular case ultimately boils down to a tabular search, we could have written one COBOL program to incorporate the whole system but then any extension or modification would have become an extremely cumbersome job. With IQL we have very powerful features and ease of functioning for formatting new reports, creating new data files or updating the old ones. Besides, if the need is felt at any stage, the IQL can also, very easily support an indexed sequential file or a data base file for the data.

The possibilities for extension are enormous, as the search can be conducted based on any field of the record as a key. Besides, with IQL any sort of aggregation, counting, comparison, copying or creation of new file structures can be done very easily. This coupled with the powerful report formatting and generation features of IQL make it a very handy tool for extension to new queries and reports.

Hence, with more user experience with this system, it can be evolved, with a reasonable amount of ease, to a stage where it becomes a truly flexible and versatile Decision Support System for several areas of activity in this Institute.

BIBLIOGRAPHY

1. Alavi, M. and Napier, H.A.; "Adaptive Design for DSS Development", Information & Management, February 1984.

2. Benbasat, Izak and Wand, Yair; "A Dialogue Generator and its Use in DSS Design", Information & Management, September-November 1982.

3. Bennett, John L.; "Analysis and Design of the User Interface for Decision Support Systems", Building Decision Support Systems : Addison-Wesley, 1983.

4. Carlson, Eric, D.; "Developing the User Interface for Decision Support Systems", Building Decision Support Systems : Addison-Wesley, 1983.

5. Carlson, Eric, D.; "An Approach for Designing Decision Support System", Data Base, Winter 1979.

6. Keen, Peter, G.W. and Morton, M.S.Scott; Decision Support Systems : An Organizational Perspective: Addison-Wesley, 1978, pp.54.

7. Lucas Jr., Henry, C.; "Computer Based Information System in Organizations"; pp.83, Science Research Associates Inc., 1973.

8. Sprague Jr., A.H.; "A Framework for Development of DSS", MIS Quarterly, December 1980.

9. Paxton, A.L. and Turner, E.J.; "The application of human factors to the needs of the novice computers user", International Journal of Man-Machine Studies; February 1984.

10. Darlington, J. Dzida and Herda, S.; "The role of excursion in Interactive System"; International Journal of Man-Machine Studies ; February 1983.

11. SWOT Analysis of Research and Development Activities: Indian Institute of Technology, Kanpur, February 1984.

APPENDIX

EXAMPLE SESSION

```
.do info

.

MJ
.Silence
Good Day ! You are running the project-information system.

You may find some delays at some stages. Please bear with us.

NOTE! Please ignore the [BREAK] and [PROCEED] which you may
      encounter during the run. These are internal commands.

WARNING! Pressing CONTROL-C at any stage would cancel whole process.

.

>If you want information
   regarding Projects- enter "PROJ"

   For information
   regarding specializations of faculty- enter "SPEC"

   To Exit - Enter "E"

@>[BREAK]PROJ

Do you want information
    Fieldwise?(Field of research)
    Departmentwise?
    Sponsorwise?
    Namewise?(Name of Faculty)
    Yearwise?(Calendar year)
    Amountwise?(Grant Amount)
    Exit? (Exit to Moniton)

@>DEP

For a full list              - enter FULL
For individual information - enter ONE

@>ONE

>thanks !
.[PROCEED]Silence _

.

>Please enter the department code eg. ME,METE,CC etc.
    (Enter "0" for a list of available departments)

@>[BREAK]0

>Thanks !
.[PROCEED]Silence
```

```
For this person  which of the following
 list compilation in progress.

 PLEASE WAIT !

 FOLLOWING DEPARTMENTS ARE THERE!

DEPARTMENT              CODE

AERONAUTICAL ENGG.      AE
CIVIL ENGG.             CE
CHEMICAL ENGG.          CHE
COMPUTER SCIENCE        CS
ELECTRICAL ENGG.        EE
MECHANICAL ENGG.        ME
METALLURGICAL ENGG.     METE
.Silence

.

>Please enter the department code eg. ME,METE,CC etc.
    (Enter "0" for a list of available departments)

@>[BREAK]ME

Within this department which of the following
information do you want?

    Sponsorwise?(for a particular sponsor)
    Namewise   ?(for a particular faculty)
    Yearwise   ?(for a particular year)
    Amountwise ?(Above a given amount)
    Fieldwise  ?(for a given research-field)
    Complete list ?
    Exit ?(Go back to MENU)

@>N

>Thanks !
.[PROCEED]Silence

As searching for the projects in  the given department.

PLEASE WAIT !

.

>Please enter the last name eg. SUNDARAJAN etc.
    (Enter "0" for a list of available Names)

@>[BREAK]SUNDARAJAN

Please enter the initials.(Each initial followed by a dot eg. B.P. or V.

@>V.
```

For this person which of the following
information do you want?

    Sponsorwise?(for a particular sponsor)
    Yearwise   ?(for a particular year)
    Amountwise ?(Above a given amount)
    Fieldwise ?(for a given research field)
    Complete list ?
    Exit ?(Go back to MENU)

@>C

>Thanks !
.[PROCEED]Silence

As searching for projects for the given faculty member.

PLEASE WAIT !

.

06/27/84 DEPARTMENTWISE LIST OF PROJECTS PAGE

DEPARTMENT: MECHANICAL ENGG.

PROJECT	SPONSOR		FACULTY MEMBER	GRANT & PERIOD
FATIGUE ANALYSIS OF ABSORBER	M/S BHARAT HEAVY PLATE AND VESSELS LTD. VISAKHAPATNAM	V.	SUNDARAJAN	8,700 1982-83
MATHEMATICAL MODELLING OF THE WHEEL SETS OF THE ROLLING STOCK	R.D.S.O. LUCKNOW	B.P. V.	SINGH SUNDARAJAN	38,000 1983-84

.Silence

.

>Do you want to retain the print file for your above query ?(YES/NO)

@>[BREAK]Y

Please give a filename of one to four characters

@>MECH

.

```
Your print file is MECH.PRI

>Thanks !
.[PROCEED]Silence

.

>Do you want to exit?(YES/NO)

@>[BREAK]N

>Okay
.[PROCEED]Silence

.

>If you want information
   regarding Projects- enter 'PROJ'

   For information
   regarding specializations of faculty- enter 'SPEC'

   To Exit - Enter 'E'

@>[BREAK]SPEC

Do you want this information
    Namewise? (Name of faculty)
    Field of Specializationwise?

@>FIELD

>Thanks !
.[PROCEED]Silence

.

>Please enter the name of the Field
    (Enter '0' for a list of available fields)
    (Enter 'E' to exit to MENU)

@>[BREAK]CAD

>Thanks !
.[PROCEED]Silence
Am searching for  names for the given research-field.

PLEASE WAIT !

.
 NO SUCH  FIELD AS   CAD   !

.@SHALL GIVE YOU A LIST OF AVAILABLE FIELDS
FOR YOUR QUERY.
.Silence

.
```

```
List compilation in progress.

PLEASE WAIT !

FOLLOWING FIELDS ARE THERE!

AERODYNAMICS
AIRCRAFT
AIRCRAFT STRUCTURE
OPTIMIZATION
O.R.

AIRCRAFT
FLUID MECHANICS
AERODYNAMICS
VIBRATION
PROPULSION
TURBINES
COMBUSTION
SPACECRAFT
FLUID MECHANICS
FLIGHT MECHANICS
AIRCRAFT STRUCTURE
OPTIMIZATION
O.R.
COMPOSITE
AIRCRAFT
FLUID MECHANICS
AERODYNAMICS
WIND TUNNEL
WIND ENERGY
COMBUSTION
PROPULSION

FLIGHT MECHANICS
AIRCRAFT
HEAT TRANSFER
MASS TRANSFER
AIRCRAFT
ANEMOMETRY
WIND ENERGY

MATERIALS
COMPOSITE
TURBINES
.Silence

>Please enter the name of the field
    (Enter '0' for a list of available fields)
    (Enter 'E' to exit to MENU)

#/ENTER/ WIND ENERGY
```

>Thanks !
.[PROCEED]Silence
Am searching for names for the given research-field

PLEASE WAIT !

06/27/84 FACULTY MEMBERS IN THE GIVEN FIELD PAGE

GIVEN FIELD: WIND ENERGY

NAME OF FACULTY	DEPARTMENT	FIELD OF SPEC.
N.L. ARORA	AERONAUTICAL ENGG.	FLUID MECHANICS AERODYNAMICS WIND TUNNEL WIND ENERGY AIRCRAFT
K. GHOSH	AERONAUTICAL ENGG.	FLUID MECHANICS WIND ENERGY AIRCRAFT

.Silence

>Do you want to retain the print file for your above query ?(YES/NO)

@>[BREAK]N

>Thanks !
.[PROCEED]Silence

>Do you want to exit?(YES/NO)

@>[BREAK]YES

>Okay
.[PROCEED]Silence

EXITING FROM THE PROJECT-INFORMATION-SYSTEM !!

Good bye